An End-to-End Knowledge Graph Fused Graph Neural Network for Accurate Protein-Protein Interactions Prediction

Jie Yang, Yapeng Li, Guoyin Wang, Zhong Chen, and Di Wu

***Abstract*—Protein-protein interactions (PPIs) are essential to understanding cellular mechanisms, signaling networks, disease processes, and drug development, as they represent the physical contacts and functional associations between proteins. Recent advances have witnessed the achievements of artificial intelligence (AI) methods aimed at predicting PPIs. However, these approaches often handle the intricate web of relationships and mechanisms among proteins, drugs, diseases, ribonucleic acid (RNA), and protein structures in a fragmented or superficial manner. This is typically due to the limitations of non-end-to-end learning frameworks, which can lead to sub-optimal feature extraction and fusion, thereby compromising the prediction accuracy. To address these deficiencies, this paper introduces a novel end-to-end learning model, the Knowledge Graph Fused Graph Neural Network (KGF-GNN). This model comprises three integral components: (1) Protein Associated Network (PAN) Construction: We begin by constructing a PAN that extensively captures the diverse relationships and mechanisms linking proteins with drugs, diseases, RNA, and protein structures. (2) Graph Neural Network for Feature Extraction: A Graph Neural Network (GNN) is then employed to distill both topological and semantic features from the PAN, alongside another GNN designed to extract topological features directly from observed PPI networks. (3) Multi-layer Perceptron for Feature Fusion: Finally, a multi-layer perceptron integrates these varied features through end-to-end learning, ensuring that the feature extraction and fusion processes are both comprehensive and optimized for PPI prediction. Extensive experiments conducted on real-world PPI datasets validate the effectiveness of our proposed KGF-GNN approach, which not only achieves high accuracy in predicting PPIs but also significantly surpasses existing state-of-the-art models. This work not only enhances our ability to predict PPIs with a higher precision but also contributes to the broader application of AI in Bioinformatics, offering profound implications for biological research and therapeutic development.**

***Index Terms*—Deep neural network, graph neural network, knowledge graph, PPI network, protein-protein interactions prediction.**

Received 20 September 2023; revised 29 June 2024; accepted 19 October 2024. Date of publication 24 October 2024; date of current version 10 December 2024. This work was supported in part by the National Science Foundation of China under Grant 62466063, Grant 62066049, Grant 62221005, Grant 61936001, and Grant 62176070, in part by the Guizhou Provincial Department of Education Colleges and Universities Science and Technology Innovation Team under Grant QJJ[2023]084, in part by Excellent Young Scientific and Technological Talents Foundation of Guizhou Province QKH-platform talent (2021) under Grant 5627, in part by Chongqing Technical Innovation and Application Development Special Project under Grant CSTB2023TIAD-KPX0037, and in part by Science and Technology Project of Zunyi under Grant ZSKRPT[2023]3. *(Corresponding author: Di Wu.)*

Jie Yang is with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the School of Physics and Electronic Science, Zunyi Normal University, Zunyi 563002, China (e-mail: yj530966074@foxmail.com).

Yapeng Li is with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: liyapeng0303@126.com).

Guoyin Wang is with the National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China (e-mail: wanggy@cqupt.edu.cn).

Zhong Chen is with the School of Computing, Southern Illinois University, Carbondale, IL 62901 USA (e-mail: zhong.chen@cs.siu.edu).

Di Wu is with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: wudi.cigit@gmail.com).

Digital Object Identifier 10.1109/TCBB.2024.3486216

I. Introduction

PROTEIN-PROTEIN interactions (PPIs) refer to the physical contact and interaction between proteins, which serve as the foundation for numerous biological processes within cells. PPIs are crucial for cellular signal transduction, metabolic pathways, gene regulation, and various other aspects [1], [2]. Unraveling protein interactions is pivotal in comprehending cellular signaling networks, disease mechanisms, and drug development [3], [4]. The mechanisms of action for many drugs involve disrupting the interactions between specific proteins. Therefore, understanding the nature and mechanisms of protein interactions contributes to the design of more effective treatment methods. The PPIs are depicted in Fig. 1. Red lines denote potential interactions between protein pairs, while blue lines signify the lack of such interactions.

The progress of computer technology has enabled artificial intelligence (AI) to play a pivotal role in the identification of protein interactions. The emergence of AI technology has addressed the limitations of traditional biological methods in identifying PPIs, which typically require lengthy experimental costs and huge computational resources. This, in turn, has improved the efficiency of PPI recognition [5]. AI methods offer the potential to analyze extensive data, model intricate interactions, and expedite the discovery of new PPIs, thereby enriching insights into cellular processes and assisting in drug discovery and development.

Currently, there are numerous AI-based models proposed for effective PPIs prediction [6], [7], [8], [9], [10], [11], [12]. Nevertheless, PPIs are intricately linked to complex relationships and mechanisms involving various factors. These factors include other proteins, drugs, diseases, ribonucleic acids, and protein structures. This indicates that PPIs are influenced not

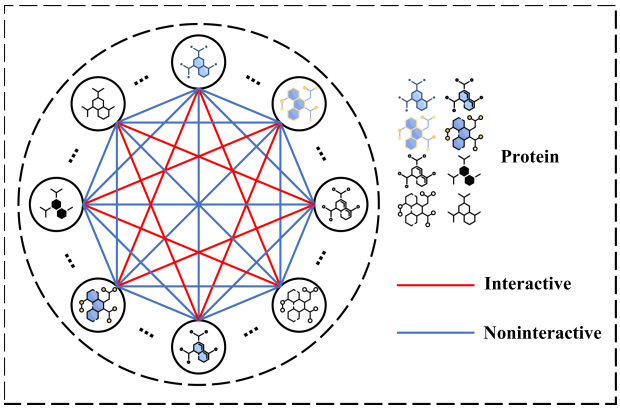

Fig. 1. An example of PPI events.

only by their own properties but also by interactions with other biomolecules and structures, forming a complex biological network. However, current research on PPIs primarily focuses on the sequences and structures of proteins themselves, often overlooking the complex relationships and mechanisms that link PPI events to proteins, drugs, diseases, ribonucleic acids, protein structures, and other factors. Furthermore, existing methods typically employ non-end-to-end models, which are often complex and prone to information loss and error accumulation, resulting in suboptimal feature extractions and fusions for prediction.

To overcome the limitations of the aforementioned methods, we propose a novel end-to-end Knowledge Graph Fused Graph Neural Network (KGF-GNN) to accurately predict PPIs. KGF-GNN consists of three essential parts. First, a protein associated network (PAN) is constructed by comprehensively exploiting protein-associated relationships and mechanisms among drugs, diseases, ribonucleic acid, protein structures, etc. A graph neural network (GNN) is then built to extract both the topological and semantic features from PAN. Second, PPI networks are constructed based on the observed interactions among proteins. Another GNN is built to extract topological features hidden in PPI networks. Third, a multi-layer perceptron is designed to fuse the extracted various features by end-to-end learning. With such designs, the feature extractions and fusions of PPIs are guaranteed to be comprehensive and nearly optimal for PPIs prediction.

The KGF-GNN model employs an end-to-end design, allowing all submodules within the model to connect seamlessly without the need for manual feature engineering or step-by-step processing. All data processing and feature extraction are automatically completed within the model. Additionally, when using the gradient descent algorithm for model optimization, the error generated can be propagated back through the model using the backpropagation algorithm, updating and optimizing the parameters of each module. This means that all modules have the ability to learn and can continuously adjust their parameters during training to enhance the overall performance of the model. This paper makes several significant contributions, which are as follows:

- Two knowledge graphs, PAN and PPI Network, are used to organize the complex relationships and mechanisms among proteins, drugs, diseases, ribonucleic acid, protein structures, and other biological entities. This enables the model to comprehensively consider the topological and semantic features among proteins and related biological entities.
- A novel end-to-end model named KGF-GNN is proposed to address the issues of information loss and error accumulation in traditional models. Through an end-to-end design, the feature extractions and fusions of PPIs are guaranteed to be comprehensive and nearly optimal for prediction.
- Our proposed model's effectiveness is validated through comprehensive experiments on two real-world PAN and PPI datasets. Comparative analysis with classic and state-of-the-art methods demonstrates the superior performance and efficacy of our approach.

Overall, these contributions highlight the significance and novelty of our method, which provide the valuable insights into the integration of PAN, the utilization of PPI Networks, and the development of the KGF-GNN model for enhanced PPI prediction. The code and dataset for the KGF-GNN model can be accessed at https://github.com/liyapeng-coder/KGF-GNN.

II. Related Work

PPI predictions can be broadly categorized into two primary groups: structure-based methods and sequence-based methods. Structure-based methods utilize information regarding protein structure similarity to predict PPIs [13]. For instance, when proteins A' and B' display structural similarity to the interacting proteins A and B correspondingly, it implies a potential interaction between A' and B' as well [14].

Protein structure is commonly visualized using three-dimensional graphics. In recent years, various variants of graph convolutional networks (GCNs) [15], [16], [17] have demonstrated successful applications in tasks involving graph-structured data. These encompass forecasts related to protein solubility [18], genome investigations [19], and the exploration of pharmaceutical compounds [20]. In the context of PPIs, a GCN-based approach is introduced to integrate positional information [21]. It fuses data from both the amino acid sequence and protein positions. Fout et al. [22] integrated 3-D structures within a GCN for precise identification of the constituent amino acids within protein interaction interfaces. For accurate prediction of interactions based solely on 3-D structural information, Baranwal et al. [9] presented a GCN-based classifier named Struct2Graph. Furthermore, a generative model outlined in [8] employs a graph-centered strategy to capture the joint distribution of the full protein sequence, long-range interactions arising from the protein structure.

In opposition to structure-oriented techniques, sequence-based approaches forego structural data and instead exploit the abundant protein sequence information derived from sequencing technologies, especially with the emergence of metagenomics [23], [24], [25]. These techniques enable PPI prediction by using amino acid sequence similarity within one species to deduce interactions in alternative species [26], [27]. Hence, the sequence-based approach primarily emphasizes the primary

structure of proteins, while disregarding their three-dimensional configuration [28].

The DPPI model [29] adopts a sequence-based PPI prediction strategy by utilizing a deep convolutional neural network with a Siamese architecture. This model integrates random projection and data augmentation methods to grasp the composition details, sequential arrangement of amino acids, and co-occurrence patterns of interactive sequence motifs within protein pairs. The PIPR approach [30] combines a deep residual recurrent convolutional neural network within the Siamese framework. Employing resilient local characteristics and contextual details from protein sequences, PIPR enhances the PPI prediction accuracy. The approach known as signed variational graph autoencoder [12] takes a graph-oriented strategy by treating the PPI Network as an undirected graph. This method combines both sequence data and graph structure to anticipate PPIs by utilizing the capabilities of variational autoencoders. The PEPPI model [31] utilizes structural similarity of protein sequences, sequence similarity, and functional association data for PPI prediction. DeepTrio [10] is another sequence-based method for PPI prediction. It employs multiple parallel convolutional neural networks alongside a masking technique, facilitating the extraction of pertinent features from protein sequences. These methods represent diverse and innovative approaches in sequence-based PPI prediction, which employ different architectures and techniques to improve the prediction accuracy.

Both structure-based methods and sequence-based methods mentioned above fall into the category of non-end-to-end models. Non-end-to-end approaches have significant drawbacks for PPI prediction owning to information loss and error accumulation. To overcome these drawbacks, we present an innovative end-to-end model called KGF-GNN. This model facilitates direct learning and prediction from input to output, which eliminates the need for intricate feature engineering. Remarkably, KGF-GNN outperforms traditional non-end-to-end models and demonstrates superior performance.

Furthermore, the structure-based and sequence-based approaches mentioned earlier concentrate on the inherent attributes of proteins themselves, without considering the possible interconnections between proteins and different biological entities like proteins, drugs, diseases, ribonucleic acid, protein structures, and more. To address this limitation, our proposed approach utilizes the PAN and PPI Network for accurate PPI prediction. Through the amalgamation of PAN and PPI Network, we acquire valuable insights into the interrelations between proteins and different biological entities, thereby boosting the overall predictive prowess. This holistic approach enables a more comprehensive understanding of PPIs by integrating various sources of information.

III. Preliminaries

A. Protein–Protein Interactions

PPIs are physical contacts of high specificity established between two or more protein molecules as a result of biochemical events steered by interactions that include electrostatic forces, hydrogen bonding and the hydrophobic effect. Many are physical contacts with molecular associations between chains that occur in a cell or in a living organism in a specific biomolecular context. A pair of proteins can interact in the following ways:

- *Physical contact:* Two proteins can have direct physical binding, such as forming dimers or multiprotein complexes. Alternatively, two proteins can interact indirectly through one or more intermediary molecules, such as adaptor proteins or scaffold proteins.
- *Co-participation in cellular processes:* Under specific cellular environments or conditions (such as specific tissue types, developmental stages, or stress conditions), two proteins co-participate in a particular cellular process (such as signal transduction, metabolic pathways, gene expression regulation, etc.).
- *Functional correlation:* The interaction between two proteins is essential for specific cellular functions, such as co-participation in a signal transduction pathway, metabolic pathway, or as part of a multi-protein complex. Alternatively, the interaction between the two proteins regulates each other's activity, stability, or localization, thereby significantly impacting cellular biological processes.

Whether interactions occur between two proteins can be checked in the STRING database [32], which aggregates, integrates, and displays PPI data.

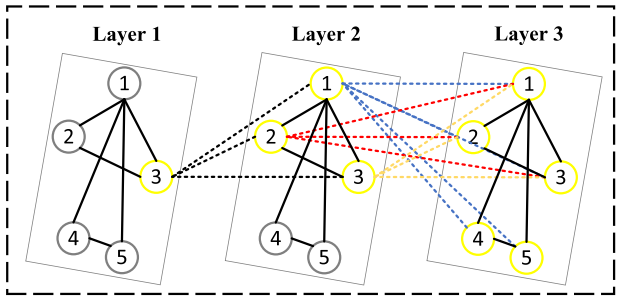

Fig. 2. Basic Ideas of Graph Neural Networks.

B. Graph Neural Network

The fundamental concept underlying GNN involves iteratively updating node features, enabling the acquisition of their representations and propagating information between nodes to capture the global context, as depicted in Fig. 2.

GNN achieve learning of node representations through the following steps:

- *Initialization of node features:* The initial features of each node are taken as input. These features are attributes of the nodes or randomly initialized vectors [33], [34].
- *Aggregation of neighbor information:* GNN aggregate information from neighboring nodes for each node. This is typically completed by computing the aggregation or weighted average of the features of the neighboring nodes. The purpose is to obtain local contextual information around the node [35], [36].
- *Information propagation:* The updated node representations are passed to the next layer. This process is iterated multiple times, with each iteration further updating the

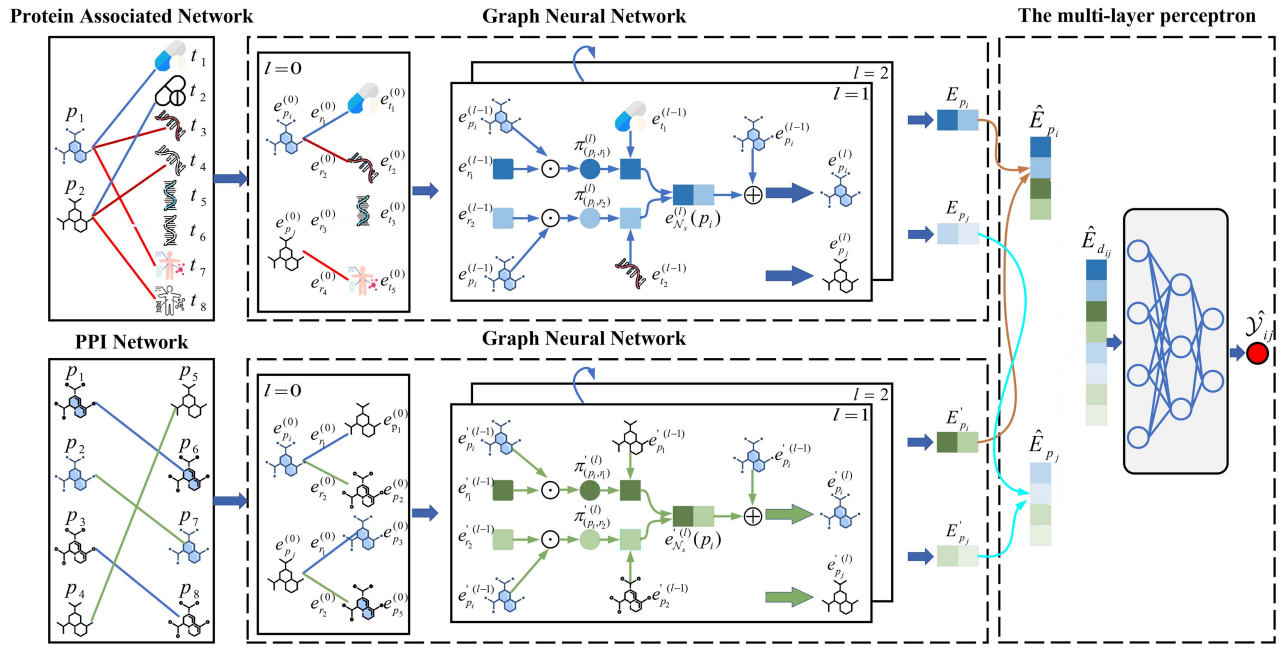

Fig. 3. The architecture of KGF-GNN.

node representations, allowing information to propagate, diffuse throughout the graph [37], [38].

- *Output prediction or features:* In the final layer of GNN, the node representations are used for specific task prediction or feature extraction. For example, in node classification tasks, the node representations input to a fully connected layer for classification [39], [40].

Through the iterative update and information propagation process, GNN is able to capture the relationships between nodes and the global structure of the graph. This allows the learning of meaningful representations for each node, which can be utilized in various graph analysis tasks, such as node classification, graph classification, and link prediction, among others.

IV. The Proposed KGF-GNN

As shown in Fig. 3, the KGF-GNN model consists of three components: two GNN parts and a multi-layer perceptron part.

First, PAN is constructed by comprehensively exploiting protein-associated relationships and mechanisms among drugs, diseases, ribonucleic acid, protein structures, etc. Then, a GNN is built to extract both the topological and semantic features from PAN.

Second, another GNN is built on the observed PPI networks to extract topological features hidden in PPIs. The second GNN part complements the first GNN by incorporating PPIs to compensate for any missing interaction relationships between proteins within the PAN. This enables the model to fully exploit the information contained in the PAN and enhances its overall performance.

Third, a multi-layer perceptron is designed to fuse two knowledge graph-based GNNs through end-to-end learning and predict the interactions between different types of proteins based on the fused feature representations. In the KGF-GNN model, the two parts of the GNN structure collaborate with each other. The first part of the GNN extracts topological and semantic features, while the second part of the GNN supplements information about interactions between proteins. This enables the KGF-GNN model to recognize more complex structures within the graph and improves the model's predictive accuracy.

In the subsequent sections, we will provide a detailed introduction to the KGF-GNN model. This comprehensive overview will shed light on how the model effectively combines GNNs and fusion techniques to predict PPIs.

A. The Graph Neural Network

The first GNN is used to generate the feature representation of proteins by extracting the topological and semantic features of the complex relationships and mechanisms between PPIs and various biological entities such as drugs, diseases, ribonucleic acids, and protein structures from the PAN. The subsequent GNN part serves to complement the first component by incorporating PPI information to compensate for any missing information within the PAN. The combination of protein feature representations generated by these two GNN parts enables the model to predict interactions between proteins more accurately.

The following briefly introduces the datasets required for the GNN parts.

Protein Associated Network: To construct the PAN, we gathered four datasets that encompass the relationships between proteins and various biological entities. These datasets include the protein-miRNA interaction dataset, protein-disease interaction dataset, protein-drug interaction dataset, and protein-LncRNA interaction dataset. The combination of these datasets forms the PAN dataset, represented as triplets in the format of

(protein, relation, entity). Here, the entity denotes one of the biological entities such as miRNA, disease, drug, or LncRNA associated with the protein. The triplets encompass four types of relations: Protein-Drug, Protein-LncRNA, Protein-miRNA, and Protein-Disease. For instance, when a triplet represents the relationship between a protein, *9606.ensp00000284981*, and an LncRNA named *nonhsat017939.2*, it would be depicted as *(9606.ensp00000284981, Protein-LncRNA, nonhsat017939.2)*. By employing this methodology, we constructed a PAN that incorporates both topological and semantic features. PAN has the capability to encompass complex relationships and mechanisms between PPIs and various biological entities such as proteins, drugs, diseases, ribonucleic acids, and protein structures.

PPI Network: Similar to the PAN in terms of format, the PPI Network contains information about the interactions between proteins. The interactions in the PPI Network are represented as triplets: *(protein, relation, protein)*. There are two types of relations: 0 and 1. A value of 0 indicates no interaction between two proteins, while a value of 1 indicates an interaction between two proteins. For example, if a protein named *9606.ensp00000267163* interacts with another protein named *9606.ensp00000267163*, it would be represented as the triplet *(9606.ensp00000267163, 1, 9606.ensp00000267163)*. The PPI Network serves as a complement to the PAN by utilizing protein interactions. This enhances the extraction of information from the PAN and improves the overall performance of the model.

Learning Process of GNN: First, we utilize the Xavier initialization method to initialize the embedding representations of proteins. The initial embedding representation of the protein knowledge graph $\mathcal{G}$ is as follows:

$$E_{\mathcal{G}} = \left[\underbrace{e_{p_1}^{(0)}, \ldots, e_{N_p}^{(0)}}_{protein\ embedding}, \underbrace{e_{r1}^{(0)}, \ldots, e_{N_r}^{(0)}}_{relation\ embedding}, \underbrace{e_{t1}^{(0)}, \cdots e_{N_t}^{(0)}}_{entity\ embedding} \right] \tag{1}$$

In formula (1), N_p, N_r, and N_{t} represent the number of proteins, relations, and entities, respectively. $e_p^{(0)} \in \mathbb{R}^d$, $e_r^{(0)} \in \mathbb{R}^d$, and $e_t^{(0)} \in \mathbb{R}^d$ represent the initialized embedding representations of proteins, relations, and entities, respectively. Here, d represents the dimension of the embedding representation.

For a protein node p_i in the knowledge graph $\mathcal{G}$, it is necessary to first randomly sample its neighboring nodes. The set of neighboring nodes of protein p_i obtained through sampling is denoted as $\mathcal{N}_s(p_i)$. The purpose of sampling neighboring nodes is to prevent model overfitting and improve the generalization performance of the model. Additionally, sampling helps to reduce the impact of outlier nodes or noisy data on the model to enhance its robustness.

Suppose there is a triple (p_i, r_{in}, t_n), where p_i represents the *i-th* protein, r_{in} represents the relation between the *i-th* protein and the *n-th* entity, and t_n represents the *n-th* entity. To aggregate the information from the edges surrounding the protein node, we combine the embedding representations of p_i and r_{in} using the following formula:

$$\pi_{(p_i, r_{in})}^{(l)} = sum\left[\left(e_{p_i}^{(l-1)} \odot e_{r_{in}}^{(l-1)}\right) W_1^{(p)} + b_1^{(p)}\right] \tag{2}$$

In the equation, $e_{r_{in}}^{(l-1)}$ represents the embedding representation of the relation between the protein p_i and the entity t_n in the $(l-1)$th layer of the GNN. $e_{p_i}^{(l-1)}$ represents the embedding representation of the protein p_i in the $(l-1)^{th}$ layer of the GNN. $W_1^{(p)}$ represents the trainable weight matrix, $b_1^{(p)}$ represents the bias vector, and p represents the number of layers in the fully connected layer. The symbol $\odot$ represents the Hadamard product.

Subsequently, based on the aforementioned aggregation of edge information, the information from protein nodes and the surrounding entity nodes is combined to form the neighborhood embedding representation $e_{N_s(p_i)}^{(l)}$ for the protein. The specific formula is as follows:

$$e_{\mathcal{N}_s(p_i)}^{(l)} = \sum_{t_n \in \mathcal{N}_s(p_i)} \pi_{(p_i, r_{in})}^{(l)} e_{t_n}^{(l-1)} \tag{3}$$

In this equation, $e_{t_n}^{(l-1)}$ represents the embedding representation of entity t_n.

Finally, in the last step, we aggregate the embedding representation $e_{p_i}^{(l-1)}$ and the neighborhood embedding representation $e_{\mathcal{N}_s(p_i)}^{(l)}$ of protein p_i using the following aggregation function:

$$E_{p_i} = e_{p_i}^{(l)} = \sigma\left(\left(e_{p_i}^{(l-1)} \oplus e_{\mathcal{N}_s(p_i)}^{(l)}\right) W_2 + b_2\right) \tag{4}$$

In this equation, $\oplus$ represents the concatenation operation. $W_2 \in \mathbb{R}^{(2d)\times d}$ is the trainable weight matrix, and b_2 represents the bias vector. The activation function σ is applied. Using the same method, the embedding representation E_{p_j} for the protein p_j can be calculated.

Furthermore, both the PAN and PPI Network have a similar format. In the second part of GNN, we extract the embedding representation of the protein from the PPI Network using the same calculation method as in the first part of GNN. Within the second part of GNN, we obtain the embedding representations E'_{p_i} and E'_{p_j} for the proteins p_i and p_j, respectively. These representations supplement the embedding representations E_{p_i} and E_{p_j} generated by the first part of GNN, thereby filling in the missing information within the PAN. Finally, PPIs are predicted and the performance of the model is enhanced by combining the protein embedding representations produced using the two GNNs.

B. The Multi-Layer Perceptron

The multi-layer perceptron is designed to fuse two knowledge graph-based GNNs through end-to-end learning and predict the interactions between different types of proteins based on the fused feature representations.

Initially, the two feature representations generated by the two GNN parts for each protein are fused together, resulting in the final embedding representation for protein p_i. The specific formula is as follows:

$$\hat{E}_{p_i} = E_{p_i} \oplus E'_{p_i} \tag{5}$$

In the equation, E_{p_i} represents the embedding generated by the first part of the GNN for protein p_i, E'_{p_i} represents the embedding

Algorithm 1: The KGF-GNN Algorithm.

Input: PAN $\mathcal{G}$(V, E), PPI Network $\mathcal{G}'(V',\ E')$
Output: Prediction score $\hat{\mathcal{Y}}_{ij}$.

initialization;
for each $v_i\ v_j \in V$,$v'_i\ v'_j \in V'$ **do**
  for each v_i,v_j,v'_i,v'_j **do**
    $\mathcal{N}_s(v) \leftarrow NeighborhoodSampling(v)$;
    $\pi_{(v)} \leftarrow sum[(e_p \odot e_r)W_1 + b_1]$;
    $e_{\mathcal{N}_s(v)} \leftarrow \sum_{t \in \mathcal{N}_s(v)} \pi_{(v)} e_t$;
    $E_v \leftarrow e_v = \sigma((e_v \oplus e_{\mathcal{N}_s(v)})W_2 + b_2)$;
  $\hat{E}_{v_i} \leftarrow E_{v_i} \oplus E_{v'_i}$;
  $\hat{E}_{v_j} \leftarrow E_{v_j} \oplus E_{v'_j}$;
  $\widehat{\mathcal{Y}}_{ij} \leftarrow \sigma\left((\hat{E}_{v_i} \oplus \hat{E}_{v_j})W_3 + b_3\right)$;
  $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$;

generated by the second part of the GNN for protein p_i, and $\hat{E}_{p_i}$ represents the final embedding representation for protein p_i. The symbol $\oplus$ represents the concatenation operation.

Similarly, following the same approach, the final embedding representation $\hat{E}_{p_j}$ for protein p_j can be obtained.

Finally, the multi-layer perceptron is employed to predict the interaction between the two proteins based on their final feature representations. The specific formula is as follows:

$$\hat{\mathcal{Y}}_{ij} = \sigma\left(\left(\hat{E}_{p_i} \oplus \hat{E}_{p_j}\right) W_3^{(q)} + b_3^{(q)}\right) \tag{6}$$

In the equation, $W_3^{(q)}$ represents the trainable weight matrix, and $b_3^{(q)}$ represents the bias vector. The symbol $\oplus$ denotes the concatenation operation and the variable q corresponds to the number of layers in the multi-layer perceptron. The activation function σ represents the sigmoid function and $\hat{\mathcal{Y}}_{ij}$ represents the final predicted score.

During model optimization, cross-entropy is employed as the loss function. The cross-entropy loss function is as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right] \tag{7}$$

In the formula, N is the total number of samples. y_i is the true label of the i-th sample, taking a value of 0 or 1. $\hat{y}_i$ is the probability that the i-th sample is predicted to belong to the positive class.

Additionally, batch normalization layers are added after each fully connected layer to accelerate the convergence speed of the model. Dropout layers and ℓ_2 regularization are incorporated to mitigate overfitting and enhance the performance of the model.

C. Algorithm Design and Complexity Analysis

The working principle of the model is described in detail in the previous section, and the pseudo-code of the model is summerized in Algorithm 1.

The time complexity of the model is expressed as $O(E \times \frac{D}{B} \times T)$, where E represents the number of epochs, D represents the size of the dataset, B represents the batch size, and T represents the time complexity of each iteration. Assuming that the dimension of the first GNN is denoted as G_1 and the dimension of the second GNN is denoted as G_2, then the time complexity of one iteration is estimated as $O(B \times (G_1 + G_2)^2)$. Consequently, the final time complexity of the model is approximated as $O(E \times D \times (G_1 + G_2)^2)$.

TABLE I
STATISTICS OF THE STUDIED DATASETS

Dataset	Subpart	Protein	Relation	Entity	Triplet
PAN	Protein-Drug	613	1	969	11107
	Protein-LncRNA	164	1	12	690
	Protein-miRNA	489	1	271	4944
	Protein-Disease	832	1	692	25087
	All	1649	4	1944	41828
PPI Network	-	1466	1	-	19237

V. EXPERIMENTS

In the subsequent experiments, we aim to answer the following research questions (RQs):

- RQ. 1. Does the proposed KGF-GNN model outperform state-of-the-art models in predicting the PPIs.
- RQ. 2. Which structures can the KGF-GNN model identify in the graph?
- RQ. 3. How do the various substructures in the model and dataset affect the performance of the model?
- RQ. 4. How do different parameters affect the model performance?

A. General Settings

Datasets: The experiment involves two datasets: the PAN and the PPI Network. These datasets are obtained from MTV-PPI [41], and we perform deletions and reconstructions on them to create a new dataset.

The PPI Network includes only positive samples, and each time the model runs, an equal number of negative samples are randomly generated. The statistical information of the datasets is presented in Table I.

The PAN dataset consists of 41,828 triplets, involving 1,649 proteins, 4 types of relations between proteins and entities, and 1,944 entities. The four relations are Protein-Drug, Protein-LncRNA, Protein-miRNA, and Protein-Disease.

The PPI Network comprises 19,237 triples, featuring 1,466 types of proteins and 1 type of relation between proteins. As a dataset of PPIs, it only includes proteins and does not incorporate entities. Since the dataset solely contains positive samples, there is only one type of relation between proteins, which is denoted as 1, indicating an interaction between the proteins.

During the training of the model, an equal number of negative samples as positive samples is randomly generated, with the relation between proteins set to 0 to indicate no interaction between them. Finally, all triples in the PAN are used as the training set. In the PPI Network, 80% of the triples are utilized as the training set, while the remaining 20% serve as the test set.

Evaluation Metrics: PPI prediction is a binary classification task, and the following evaluation metrics are used to measure the performance of the model: Accuracy (Acc), Recall (Rec),

Precision (Pre), Matthews Correlation Coefficient(MCC), Area Under Curve (AUC), and Area Under Precision-Recall Curve (AUPR).

In the following formula, TP (True Positive) signifies correct positive classifications, TN (True Negative) represents correct negative classifications, FP (False Positive) indicates erroneous positive classifications, and FN (False Negative) stands for incorrect negative classifications.

Acc is a metric that gauges the correctness of predictions across the entire dataset, thus assessing the overall model performance. However, in the case of imbalanced data, it may give misleading results. The specific formula for Acc is as follows:

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{8}$$

Rec quantifies a model's aptitude to detect positive instances, specifically the ratio of true positives accurately predicted as positive. A higher recall value indicates a model's enhanced capability in recognizing positive instances. The specific formula is defined as follows:

$$Rec = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{9}$$

Pre evaluates the ratio of predicted positive samples that truly hold positive status. A greater precision signifies an improved accuracy of the model when predicting positive samples. The specific formula is defined as follows:

$$Pre = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

The MCC evaluation metric is suitable for imbalanced datasets. It combines the four types of prediction results (TP, TN, FP, FN) to provide a comprehensive measure. The specific formula is defined as follows:

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{11}$$

AUC quantifies the model's capacity to predict samples across various thresholds, with a higher value indicating improved predictive prowess. Its calculation method is as follows:

First, by adjusting the threshold of the model, a series of predicted results and corresponding true labels are obtained. Then, the TPR (True Positive Rate) and FPR (False Positive Rate) are calculated at each threshold. TPR and FPR are defined as follows:

$$TPR = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{12}$$

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{13}$$

Then, according to the obtained series of TPR and FPR, FPR is used as the abscissa and TPR is used as the ordinate to draw the ROC curve, and AUC is obtained by calculating the area under the ROC curve.

AUPR is different from AUC in that AUPR is a more suitable evaluation metric for imbalanced data sets, because Acc and AUC may be inaccurate in imbalanced data sets. It is calculated as follows:

For different threshold values, calculate the corresponding Pre and Rec, and then plot them as a Precision-Recall curve. Compute the area under the Precision-Recall curve, referred to as AUPR.

Baselines: We conducted a comprehensive comparison between the proposed KGF-GNN model and seven state-of-the-art related models. The specific details of the compared models are presented in Table II.

Throughout the comparison, we assessed each model's efficacy in PPI prediction by employing a range of metrics. The results provide insights into the strengths and weaknesses of different approaches and highlight the advantages of our proposed KGF-GNN model.

Implementation Detail: A rigorous evaluation was conducted by performing five-fold cross-validation on all models. The final results were obtained by averaging the outcomes of the five experiments. Additionally, the standard deviation across the five experiments was calculated to assess the stability of the models.

Since the PPI Network only contains positive samples, it was necessary to generate an equal number of negative samples randomly. The combined dataset, consisting of both positive and negative samples, was used for the experiments. In the five-fold cross-validation, the PPI Network was divided randomly into five equal parts. For each experiment, the PAN dataset and 80% of the PPI Network were utilized as the training set, while the remaining 20% of the PPI Network served as the test set.

To mitigate any potential asymmetry in the PPI Network, we reversed the order of two proteins in the triples of the PPI Network training set and added them to the end of the PPI Network training set. This ensured that the dataset maintained balance and accounted for potential biases during model training and evaluation.

System Configuration: The code is written in Python3.9 and pytorch, pandas, numpy, sklearn, etc are also used. The experiments were performed on a computer with i5-8300H CPU and GTX1050 GPU.

B. Performance Comparison (RQ. 1)

We compared our proposed model, KGF-GNN, with state-of-the-art models, and the specific comparison results are shown in Table III. To evaluate the performance of the model across various aspects, we employ multiple evaluation metrics including Acc, Rec, Pre, MCC, AUC, and AUPR. These metrics provide a comprehensive assessment of the model's performance. Furthermore, to better analyze these experimental results, we conducted statistical analyses of the data using measures such as win/loss ratio, Wilcoxon signed-ranks test, and Friedman test.

The win/loss ratio refers to the number of evaluation metrics in which the KGF-GNN model outperforms or underperforms compared to the baseline models. The Wilcoxon signed-ranks test is a nonparametric pairwise comparison method. It checks whether KGF-GNN has significantly higher prediction accuracy than each comparison model by the level of significance p-value. The Friedman test is to compare the performance of multiple models on multiple datasets simultaneously by the F-rank value. A lower F-rank value denotes a higher prediction accuracy.

TABLE II
SUMMARY OF COMPARISON MODELS

Model	Description
WSRC_GE [42]	The method presented in this study includes feature extraction from protein sequences and the introduction of an innovative weighted sparse representation-based classifier for PPI prediction. (BMC Bioinformatics 2016)
LR_PPI [43]	The model presented in this study is a sequence-based PPI prediction model. It employs a stacked autoencoder to encode protein sequences and make subsequent PPI predictions. (BMC Bioinformatics 2017)
DPPI [29]	The proposed model in this study is a sequence-based PPI prediction model. It employs a combination of CNN, as well as techniques involving random projection and data augmentation, for PPI prediction. (Bioinformatics 2018)
PIPR [30]	The model presented in this research integrates a deep residual recurrent convolutional neural network (RRCNN) within the Siamese architecture. It utilizes protein sequences to predict PPIs. (Bioinformatics 2019)
LPPI [44]	The approach utilized in this study involves reconstructing a weighted network based on protein basic information. Subsequently, the protein network representation is learned using DeepWalk, followed by PPI sample classification using Logistic Regression (LR). (Front Genet 2021)
MDNN [45]	The proposed method adopts a multimodal approach to integrate information from knowledge graphs and heterogeneous features, aiming to predict interactions between drugs. (IJCAI 2021)
MTV-PPI [41]	The method employs K-mer and LINE techniques to extract features from protein sequences and heterogeneous molecular networks, respectively. Subsequently, a random forest classifier is utilized for PPI prediction. (BMC Bioinformatics 2022)
TAGPPI [46]	This study proposes a protein sequence-based model that performs convolution operations on protein sequences and extracts features from contact maps for PPI prediction. (Briefings in Bioinformatics 2022)
HNSPPI [47]	This study proposes a hybrid supervised learning model that integrates amino acid sequence information and PPI network connectivity features to comprehensively characterize protein features for PPI prediction. (Briefings in Bioinformatics 2023)
EResCNN [5]	This study proposes a model based on an ensemble residual convolutional neural network, which extracts and integrates protein features and integrates multiple classifiers for PPI prediction. (Elsevier 2023)
KGF-GNN	Our approach combines the PAN and PPI Network using a GNN approach to predict PPIs in an end-to-end manner. (Ours)

TABLE III
MODEL COMPARISON RESULTS

Model	Acc	Rec	Pre	AUC	AUPR	MCC	Win/Loss	*p*-value	F-rank
LR_PPI	0.7717±0.0066	0.7551±0.0090	0.7329±0.0092	0.8482±0.0060	0.8411±0.0058	0.6027±0.0089	5/0	0.0156	10.00
DPPI	0.8007±0.0087	0.7623±0.0099	0.7677±0.0090	0.8726±0.0076	0.8903±0.0078	0.6334±0.0097	5/0	0.0156	8.58
WSRC_GE	0.8225±0.0105	0.7623±0.0097	0.7987±0.0123	0.9022±0.0089	0.8975±0.0086	0.6996±0.0118	5/0	0.0156	6.42
LPPI	0.8062±0.0116	**0.9275±0.0124**	0.7232±0.0103	0.8424±0.0173	0.8022±0.0154	0.6779±0.01587	4/1	0.0313	8.00
PIPR	0.7536±0.0090	0.7678±0.0100	0.7456±0.0098	0.8331±0.0094	0.8246±0.0096	0.6623±0.0097	5/0	0.0156	9.50
MTV-PPI	0.8655±0.0050	0.8249±0.0085	0.8979±0.0088	0.9301±0.0050	0.9308±0.0045	0.6838±0.0079	5/0	0.0156	3.83
MDNN	0.8337±0.0027	0.8607±0.0082	0.8167±0.0071	0.8998±0.0009	0.8823±0.0046	0.6595±0.0063	5/0	0.0156	6.83
HNSPPI	0.9031±0.0033	0.9040±0.0050	0.8797±0.0063	0.9031±0.0034	0.9233±0.0030	0.8078±0.0065	5/0	0.0156	2.83
TAGPPI	0.8739±0.0046	0.9015±0.0053	0.8544±0.0042	0.8738±0.0037	0.9026±0.0057	0.7488±0.0049	5/0	0.0156	4.33
EResCNN	0.8480±0.0024	0.8613±0.0056	0.8390±0.0018	0.9223±0.0022	0.9218±0.0031	0.6963±0.0050	5/0	0.0156	4.50
KGF-GNN (ours)	**0.9065±0.0033**	0.9087±0.0041	**0.9048±0.0036**	**0.9548±0.0017**	**0.9460±0.0033**	**0.8207±0.0024**	**49/1**	**-**	**1.17**

Additionally, we compared the runtime of our proposed KGF-GNN model with state-of-the-art models. Under the same experimental conditions and dataset, we measured the runtime of each model. The experimental results are shown in Fig. 4.

Based on the data presented in the Table III and Fig. 4, it is evident that the KGF-GNN model exhibits excellent performance across all evaluation metrics. A detailed analysis is provided below:

- In the 50 comparisons with other models based on evaluation metrics, the KGF-GNN model has achieved an impressive 49 wins and only 1 loss. The KGF-GNN model has a Rec metric that does not exceed that of the LPPI model. However, it outperforms the LPPI model by at least 0.1 on the other four evaluation metrics. This indicates that the KGF-GNN model consistently outperforms other models, demonstrating its superior accuracy in predicting protein-protein interactions.
- Among all the models, the KGF-GNN model stands out with the highest F-rank, reaffirming its superior predictive performance. This result solidifies the KGF-GNN

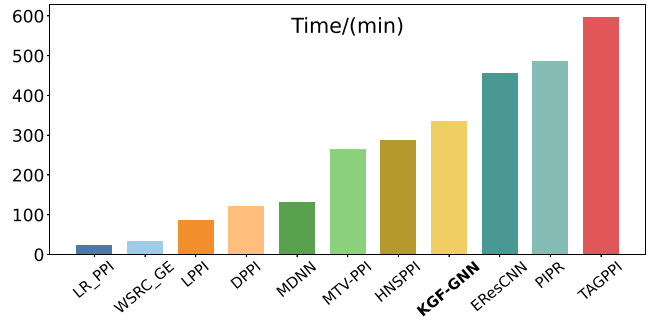

Fig. 4. Running Times of Different Models.

model's position as the top performer among the compared models.

- All the obtained p-values are below the 0.05 significance level, providing strong evidence that the prediction performance of KGF-GNN surpasses that of all the compared models with a significance level of 0.05. This significant difference further highlights the superiority of the KGF-GNN model in terms of predictive capabilities.
- Furthermore, it is worthy noting that the KGF-GNN model demonstrates a lower standard deviation compared to the majority of other models. This finding suggests that the KGF-GNN model exhibits remarkable stability in its predictions, indicating consistent performance across different experiments or datasets.
- In the comparison of model runtimes, some models, having been proposed earlier, feature simpler computational processes and thus have shorter runtimes compared to the KGF-GNN model. However, the prediction accuracy of the KGF-GNN model is significantly superior to these models. Therefore, to achieve better prediction results, the runtime of the KGF-GNN model is within an acceptable range.

The KGF-GNN model's ability to improve the accuracy of PPI prediction is attributed to several factors. On the one hand, it benefits from being an end-to-end model, eliminating information loss and error accumulation that can occur in complex feature engineering. On the other hand, the model leverages the collaboration between its two GNN components. One part of the GNN extracts the topological and semantic features of the complex relationships and mechanisms between PPIs and various biological entities such as drugs, diseases, ribonucleic acids, and protein structures, while the other part supplements interaction information between proteins. This synergy allows the KGF-GNN model to recognize more complex structures within the graph, thereby boosting the predictive accuracy.

C. Case Study (RQ. 2)

In this section of the experiment, the main focus is on investigating which graph structures the KGF-GNN model can identify. Since the MDNN model shares a similar structure with the KGF-GNN model, it is used as a comparative model for this experiment. First, the second-order neighborhood graphs in the PPI network for two proteins correctly predicted by the MDNN model are drawn, as shown in the Fig. 5. Then, the second-order neighborhood graphs in the PPI network for two proteins, which were predicted correctly by the KGF-GNN model but incorrectly by the MDNN model, are drawn, as shown in the Fig. 6.

From the comparison of the graph structures identified by the two models, it can be observed that the graph structures recognized by the KGF-GNN model are more complex than those recognized by the MDNN model. In the graph structures identified by the KGF-GNN model, there are more neighboring nodes and connections between neighboring nodes, while in the graph structures identified by the MDNN model, there are fewer neighboring nodes and fewer connections between them.

The reason why the KGF-GNN model is able to recognize more complex structures is closely tied to its two-part GNN architecture, whereas the MDNN model only has a single GNN structure, resulting in a weaker ability to identify complex graph structures. The MDNN model has only one part of the GNN structure, while the other part uses the Jaccard similarity to calculate the first-order similarity between nodes, and it does not participate in model training. Therefore, its ability to recognize complex graph structures is weak. Experimental results indicate that the two parts of the GNN structure in the KGF-GNN model collaborate with each other: the first part of the GNN extracts the topological and semantic features of the complex relationships and mechanisms between PPIs and various biological entities such as drugs, diseases, ribonucleic acids, and protein structures, while the second part of the GNN supplements information about interactions between proteins. This synergy enables the KGF-GNN model to identify more complex structures within the graph, thus enhancing the model's predictive accuracy.

D. Ablation Study (RQ. 3)

In order to investigate the impact of different components in the model and dataset on the model's performance, we conducted an ablation study. Ablation experiments involve selectively removing or excluding specific factors to assess their contribution to the model's performance. By isolating and analyzing these factors, insights into the underlying mechanisms driving the experimental results can be gained.

This section presents two main ablation experiments. The first experiment focuses on examining the effects of two components within the GNN on the model's performance. The second experiment aims to evaluate how different substructures within the PAN influence the model's performance. These experiments provide valuable insights into understanding the role of specific factors in the model's performance. In the following experiments, the results are the averages from five-fold cross-validation.

1) Ablation Experiments for the Model: In the first ablation experiment, we conducted a comparative analysis by separating the two GNN components in the model and evaluating their performance individually. The experimental results are presented in Fig. 7. From the figure, it is evident that the GNN model with the PAN component exhibits lower performance compared to the GNN model with the PPI Network component. Across all evaluation metrics except for the MCC metric, the GNN model with the PAN compo-nent has an average decrease of approximately 0.02in performance compared to the GNN model with

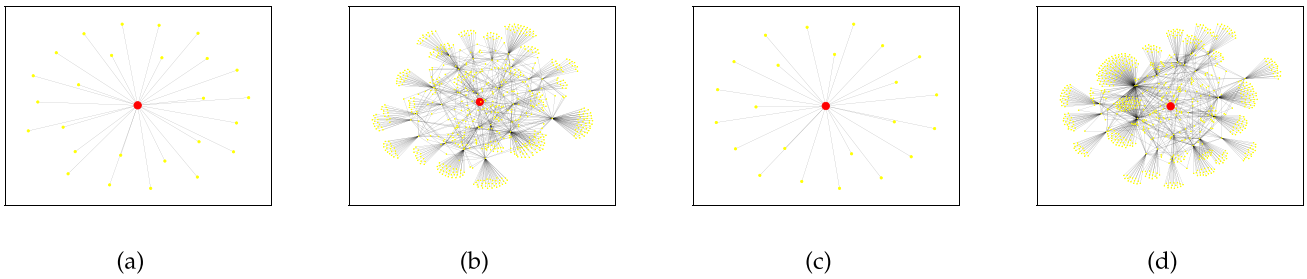

Fig. 5. The graph structures identified by the MDNN model. (a) The first-order neighbor graph of the first node in the prediction sample. (b) The second-order neighbor graph of the first node in the prediction sample. (c) The first-order neighbor graph of the second node in the prediction sample. (d) The second-order neighbor graph of the second node in the prediction sample.

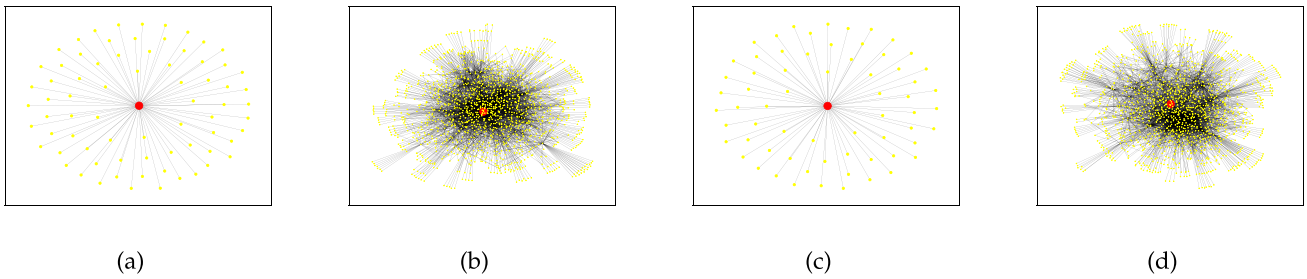

Fig. 6. The graph structures identified by the KGF-GNN model. (a) The first-order neighbor graph of the first node in the prediction sample. (b) The second-order neighbor graph of the first node in the prediction sample. (c) The first-order neighbor graph of the second node in the prediction sample. (d) The second-order neighbor graph of the second node in the prediction sample.

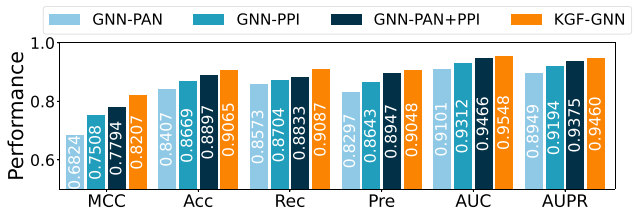

Fig. 7. Results of ablation experiments for the model.

the PPI Network component. In terms of the MCC evaluation metric, GNN-PPI improved by approximately 0.07 compared to GNN-PAN. Additionally, the performance of each individual model, after separating the two components, is not superior to the performance of the combined model with both components. The combined model with both components demonstrates an improvement of approximately 0.03 across all evaluation metrics compared to the best-performing individual model. These results indicate that the combination of both components contributes to the overall enhancement of the model's performance. The GNN model with the PPI Network component compensates for the absence of PPI information in the PAN. This enables more effective extraction of topological and semantic features of the complex relationships and mechanisms between PPIs and various biological entities such as drugs, diseases, ribonucleic acids, and protein structures from the graph by the GNN model with the PAN component, making an improved performance of the model.

Additionally, the KGF-GNN uses two GNN components to separately extract protein information from the PAN and PPI networks. To validate the necessity of this two-track architecture, we conducted a fusion experiment on the two GNN components. We merged the PAN and PPI networks into a single network and used one GNN module to extract protein information, embedding the extracted protein features for subsequent PPI prediction. As shown in Fig. 7, the performance of the GNN-PAN+PPI model surpasses that of the single GNN module extracting features from a single protein network but is inferior to the model with two GNN modules extracting features from two protein networks. Due to the differences between the interaction information of PPIs and the interaction information between proteins and other biological entities, merging these two types of information into a single protein network can mislead the GNN. This results in less precise protein feature embeddings, thereby reducing the model's accuracy. The KGF-GNN model uses two separate GNNs to extract PPI information and protein interactions with other biological entities. Each GNN extracts a single type of protein information, resulting in more precise protein feature embeddings and thereby enhancing the model's accuracy.

2) Ablation Experiments for Dataset PAN: In the second ablation experiment, we focused on the entities within the PAN, specifically the categories of Drug, LncRNA, miRNA, and Disease. We conducted individual evaluations using the GNN-PAN model to examine the impact of each entity category on the model's performance. The experimental results are presented in Fig. 8. From the figure, It is noticeable that each entity category within the PAN dataset contributes similarly to the model's performance. However, Disease entities exhibit a slightly higher contribution compared to the other three entity types, suggesting a potentially stronger association between Disease entities and proteins. Additionally, integrating all four entity types to

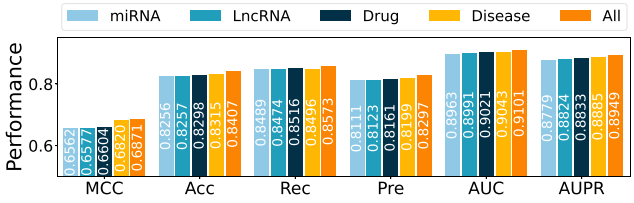

Fig. 8. Results of ablation experiments for PAN.

construct a complete PAN leads to an improvement of approximately 0.01in the evaluation metrics of the GNN-PAN model. This finding indicates that the incorporation of all entities provides the model with richer topological and semantic features, ultimately enhancing its performance.

E. Parameter Sensitivity Analysis (RQ. 4)

In this section, we will analyze the impact of hyperparameters on the performance of the model. The model incorporates several hyperparameters, and we will primarily focus on the coefficients of the ℓ_2 regularization term, the dimensions of the embedding representations in the two GNN components, and the size of the neighborhood sampling. Through examining the variations in these parameters and their corresponding results, insights into the underlying mechanisms of the model can be gained, allowing inference about how it processes the data. This analysis helps us understand the model's decision-making process and outcomes, and assists in determining the optimal range of parameter values. The experimental results are depicted in Fig. 9.

1) The Impact of the ℓ_2 Regularization Coefficient on Model Performance: When the coefficient of the ℓ_2 regularization term is excessively large, it constrains the model to adopt simpler functional forms, which may limit its ability to capture the intricate relationships within the data. This can result in a decline in model performance. Conversely, when the coefficient is too small, the model becomes more susceptible to overfitting the training data. Overfitting can yield impressive performance on the training set but poor generalization to unseen data. From the figure, it is evident that the model achieves optimal performance when the ℓ_2 regularization coefficient is set to 0.01.

2) The Impact of the Dimensionality of Embedding Representations on Model Performance: A smaller dimension for embedding representations can potentially result in information loss, as the model may struggle to capture the intricate patterns and structures within the data or fully learn its representation. This can lead to a decreased model performance. From the figure, it is evident that as the dimension of the GNN embedding representations increases, the model's performance improves. However, it is important to note that a larger dimension also increases the number of model parameters and computational complexity, particularly when dealing with large-scale datasets. This requires additional computational resources and time for training. Therefore, it is crucial to carefully consider the trade-off between model performance and computational complexity, aiming for a tradeoff that meets the specific requirements of the task.

3) The Impact of Neighborhood Sampling Size on Model Performance: A large neighborhood sampling size may lead the model to excessively focus on the global structure, potentially overlooking the important local neighborhood information of the nodes. In certain cases, this local neighborhood information plays a crucial role in node representation and prediction, and an excessively large sampling size can overshadow this information, resulting in a decline in model performance. On the other hand, a small neighborhood sampling size may result in information loss and an incomplete understanding of the relationships and information surrounding the nodes, limiting the model's learning capacity. Therefore, it is essential to carefully choose an appropriate neighborhood sampling size that balances the local and global perspectives.

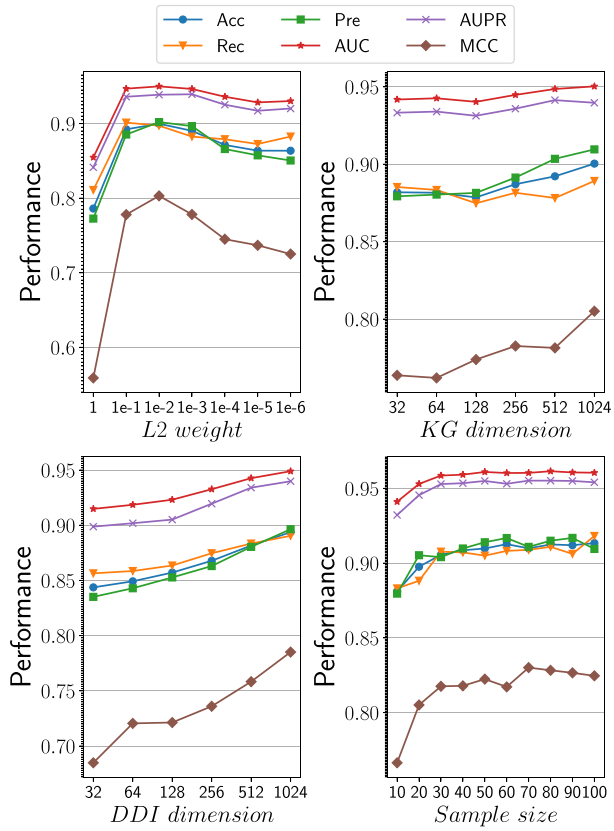

Fig. 9. The influence of different parameters on model performance.

From the figure, the predicted results exhibit a stronger correlation with the global structure of the dataset. As the neighborhood sampling size increases, the model's performance improves. However, it is important to consider the computational resources and time required for training, as a large sampling size increases the computational complexity of the model. Therefore, finding the optimal tradeoff between capturing local neighborhood information and considering the global structure is crucial for achieving the best model performance. In our case, from Fig. 9, we can select the neighborhood sampling size to be 60, better balancing these two scenarios.

VI. Conclusion

In this paper, we introduce the Knowledge Graph Fused Graph Neural Network (KGF-GNN), a novel model designed to predict protein-protein interactions (PPIs) with high accuracy. Our

model employs an end-to-end learning framework that seamlessly integrates feature extraction and predictive modeling. This approach not only captures the complex interrelationships among various biological entities-including drugs, diseases, ribonucleic acids, and protein structures-but also maintains model simplicity and interpretability. The KGF-GNN consists of two distinct Graph Neural Network (GNN) architectures complemented by a multi-layer perceptron. The first GNN is tasked with extracting both topological and semantic features from the Protein Associated Network (PAN), generating comprehensive protein feature representations. The second GNN enhances these features by addressing potential gaps in the PAN, specifically by identifying and integrating missing interactions among proteins. The multi-layer perceptron is crucial for synthesizing the outputs of both GNNs, effectively fusing the diverse protein features to predict interactions across various protein types. This holistic use of information significantly boosts the model's performance.

We rigorously evaluated the KGF-GNN model against contemporary state-of-the-art models through comparative experiments. The results clearly demonstrate our model's superiority, as it consistently outperforms others in predictive accuracy. Unlike other models, the KGF-GNN offers a streamlined, end-to-end solution that simplifies the predictive process and enhances performance, making it a pragmatic and powerful tool for PPI prediction.

As part of future work, we plan to augment our dataset with additional protein types and integrate attention mechanisms to refine the interaction between the two GNN components, such as the data adopted by the PEPPI model [31]. We will also explore the application of the KGF-GNN model across other research domains, expanding its utility and impact.

REFERENCES

[1] C. Chen, Q. Zhang, Q. Ma, and B. Yu, "Lightgbm-PPI: Predicting protein-protein interactions through lightgbm with multi-information fusion," *Chemometrics Intell. Lab. Syst.*, vol. 191, pp. 54–64, 2019.

[2] X. Li, P. Han, G. Wang, W. Chen, S. Wang, and T. Song, "SDNN-PPI: Self-attention with deep neural network effect on protein-protein interaction prediction," *BMC Genomic.*, vol. 23, no. 1, 2022, Art. no. 474.

[3] D. F. Burke et al., "Towards a structurally resolved human protein interaction network," *Nat. Struct. Mol. Biol.*, vol. 30, no. 2, pp. 216–225, 2023.

[4] A. Dhakal, C. McKay, J. J. Tanner, and J. Cheng, "Artificial intelligence in the prediction of protein–ligand interactions: Recent advances and future directions," *Brief. Bioinf.*, vol. 23, no. 1, 2022, Art. no. bbab476.

[5] H. Gao, C. Chen, S. Li, C. Wang, W. Zhou, and B. Yu, "Prediction of protein-protein interactions based on ensemble residual convolutional neural network," *Comput. Biol. Med.*, vol. 152, 2023, Art. no. 106471.

[6] P. Bryant, G. Pozzati, and A. Elofsson, "Improved prediction of protein-protein interactions using alphafold2," *Nat. Commun.*, vol. 13, no. 1, 2022, Art. no. 1265.

[7] X. Luo, L. Wang, P. Hu, and L. Hu, "Predicting protein-protein interactions using sequence and network information via variational graph autoencoder," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 5, pp. 3182–3194, Sep./Oct. 2023.

[8] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola, "Generative models for graph-based protein design," in *PRoc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15820–15831.

[9] M. Baranwal et al., "Struct2Graph: A graph attention network for structure based predictions of protein–protein interactions," *BMC Bioinf.*, vol. 23, no. 1, 2022, Art. no. 370.

[10] X. Hu, C. Feng, Y. Zhou, A. Harrison, and M. Chen, "DeepTrio: A ternary prediction system for protein–protein interaction using mask multiple parallel convolutional neural networks," *Bioinformatics*, vol. 38, no. 3, pp. 694–702, 2022.

[11] J. Wu, E. Paquet, H. L. Viktor, and W. Michalowski, "Paying attention: Using a siamese pyramid network for the prediction of protein-protein interactions with folding and self-binding primary sequences," in *Proc. 2021 IEEE Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.

[12] F. Yang, K. Fan, D. Song, and H. Lin, "Graph-based prediction of protein-protein interactions with attributed signed graph embedding," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–16, 2020.

[13] J. Hoskins, S. Lovell, and T. L. Blundell, "An algorithm for predicting protein–protein interaction sites: Abnormally exposed amino acid residues and secondary structure elements," *Protein Sci.*, vol. 15, no. 5, pp. 1017–1029, 2006.

[14] T.-L. Shi, Y.-X. Li, Y.-D. Cai, and K.-C. Chou, "Computational methods for protein-protein interaction and their application," *Curr. Protein Peptide Sci.*, vol. 6, no. 5, pp. 443–449, 2005.

[15] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[17] K. Madi, E. Paquet, and H. Kheddouci, "New graph distance for deformable 3d objects recognition based on triangle-stars decomposition," *Pattern Recognit.*, vol. 90, pp. 297–307, 2019.

[18] J. Chen, S. Zheng, H. Zhao, and Y. Yang, "Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map," *J. Cheminformatics*, vol. 13, no. 1, pp. 1–10, 2021.

[19] J. Rao, X. Zhou, Y. Lu, H. Zhao, and Y. Yang, "Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks," *Iscience*, vol. 24, no. 5, 2021, Art. no. 102393.

[20] Y. Song, S. Zheng, Z. Niu, Z.-H. Fu, Y. Lu, and Y. Yang, "Communicative representation learning on attributed molecular graphs," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 2831–2838.

[21] L. Liu et al., "Combining sequence and network information to enhance protein–protein interaction prediction," *BMC Bioinf.*, vol. 21, no. 16, pp. 1–13, 2020.

[22] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6533–6542.

[23] "Uniprot: The universal protein knowledgebase in 2021," *Nucleic acids Res.*, vol. 49, no. D1, pp. D480–D489, 2021.

[24] P. Hugenholtz and G. W. Tyson, "Metagenomics," *Nature*, vol. 455, no. 7212, pp. 481–483, 2008.

[25] A. Cumberworth, G. Lamour, M. M. Babu, and J. Gsponer, "Promiscuity as a functional trait: Intrinsically disordered regions as central players of interactomes," *Biochem. J.*, vol. 454, no. 3, pp. 361–369, 2013.

[26] T. Wang, L. Li, Y.-A. Huang, H. Zhang, Y. Ma, and X. Zhou, "Prediction of protein-protein interactions from amino acid sequences based on continuous and discrete wavelet transform features," *Molecules*, vol. 23, no. 4, 2018, Art. no. 823.

[27] J. R. Bock and D. A. Gough, "Predicting protein–protein interactions from primary structure," *Bioinformatics*, vol. 17, no. 5, pp. 455–460, 2001.

[28] K.-C. Chou and G. M. Maggiora, "Domain structural class prediction," *Protein Eng.*, vol. 11, no. 7, pp. 523–538, 1998.

[29] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, "Predicting protein–protein interactions through sequence-based deep learning," *Bioinformatics*, vol. 34, no. 17, pp. i802–i810, 2018.

[30] M. Chen et al., "Multifaceted protein–protein interaction prediction based on siamese residual RCNN," *Bioinformatics*, vol. 35, no. 14, pp. i305–i314, 2019.

[31] E. W. Bell, J. H. Schwartz, P. L. Freddolino, and Y. Zhang, "PEPPI: Whole-proteome protein-protein interaction prediction through structure and sequence similarity, functional association, and machine learning," *J. Mol. Biol.*, vol. 434, no. 11, 2022, Art. no. 167530.

[32] D. Szklarczyk et al., "The string database in 2023: Protein–protein association networks and functional enrichment analyses for any sequenced genome of interest," *Nucleic Acids Res.*, vol. 51, no. D1, pp. D638–D646, 2023.

[33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[34] A. Pareja et al., "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5363–5370.

[35] S. M. Kazemi et al., "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 2648–2720, 2020.

[36] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey," *IEEE Access*, vol. 9, pp. 79 143–79 168, 2021.

[37] J. Huang, H. Shen, L. Hou, and X. Cheng, "SDGNN: Learning node representation for signed directed networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 196–203.
[38] M. Gao, L. Chen, X. He, and A. Zhou, "BiNE: Bipartite network embedding," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 715–724.
[39] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11983–11993.
[40] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1358–1368.
[41] X.-R. Su, L. Hu, Z.-H. You, P.-W. Hu, and B.-W. Zhao, "Multi-view heterogeneous molecular network representation learning for protein–protein interaction prediction," *BMC Bioinf.*, vol. 23, no. 1, 2022, Art. no. 234.
[42] Y.-A. Huang, Z.-H. You, X. Chen, K. Chan, and X. Luo, "Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding," *BMC Bioinf.*, vol. 17, pp. 1–11, 2016.
[43] T. Sun, B. Zhou, L. Lai, and J. Pei, "Sequence-based prediction of protein protein interaction using a deep-learning algorithm," *BMC Bioinf.*, vol. 18, pp. 1–8, 2017.
[44] X.-R. Su, Z.-H. You, L. Hu, Y.-A. Huang, Y. Wang, and H.-C. Yi, "An efficient computational model for large-scale prediction of protein–protein interactions based on accurate and scalable graph embedding," *Front. Genet.*, vol. 12, 2021, Art. no. 635451.
[45] T. Lyu, J. Gao, L. Tian, Z. Li, P. Zhang, and J. Zhang, "MDNN: A multimodal deep neural network for predicting drug-drug interaction events," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3536–3542.
[46] B. Song, X. Luo, X. Luo, Y. Liu, Z. Niu, and X. Zeng, "Learning spatial structures of proteins improves protein–protein interaction prediction," *Brief. Bioinf.*, vol. 23, no. 2, 2022, Art. no. bbab558.
[47] S. Xie et al., "HNSPPI: A hybrid computational model combing network and sequence information for predicting protein–protein interaction," *Brief. Bioinf.*, vol. 24, no. 5, 2023, Art. no. bbad261.

Jie Yang received the PhD degree in computer science and technology from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2019. He is currently a professor with Zunyi Normal University, Zunyi, China, and a masters supervisor with the Chongqing University of Posts and Telecommunications. He has more than 50 publications, including *IEEE Transactions on Fuzzy Systems*, *Information Sciences*, *Knowledge-Based Systems*, etc. His research interests include data mining, machine learning, three-way decisions, and rough sets.

Yapeng Li received the BS degree in computer science from Qingdao Agricultural University, Qingdao, China, in 2022. He is currently working toward the MS degree in computer technology with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include graph neural network and machine learning.

Guoyin Wang (Senior Member, IEEE) received the BS, MS, and PhD degrees from Xi'an Jiaotong University, Xian, China, in 1992, 1994, and 1996, respectively. He worked with the University of North Texas, and the University of Regina, Canada, as a visiting scholar during 1998–1999. He had worked with the Chongqing University of Posts and Telecommunications during 1996–2024, where he was a professor, the vice-president of the University, the director of the Chongqing Key Laboratory of Computational Intelligence, the director of the Key Laboratory of Cyberspace Big Data Intelligent Security, Ministry of Education, and the director of the Sichuan-Chongqing Joint Key Laboratory of Digital Economy Intelligence and Security. He was the director of the Institute of Electronic Information Technology, Chongqing Institute of Green and Intelligent Technology, CAS, China, 2011–2017. He has been serving as the president of Chongqing Normal University since June 2024. He is the author of more than 10 books, the editor of dozens of proceedings of international and national conferences and has more than 300 reviewed research publications. His research interests include rough sets, granular computing, machine learning, knowledge technology, data mining, neural network, cognitive computing, etc. He was the President of International Rough Set Society (IRSS) 2014–2017, and a council member of the China Computer Federation (CCF) 2008–2023. He is a vice-president of the Chinese Association for Artificial Intelligence (CAAI). He is a fellow of IRSS, CAAI and CCF.

Zhong Chen (Member, IEEE) received the PhD degree from the Wuhan University of Technology, China in 2015. He is currently an assistant professor with the School of Computing, Southern Illinois University, Carbondale, IL, USA. His main research interests include data-centric AI, deep learning, machine learning, data mining, online learning, Bioinformatics, and medical physics. He has published 36 peer-reviewed articles in scientific journals and conferences such as *Physical Review Letters*, *Image and Vision Computing*, *Journal of Machine Learning and Cybernetics*, *Knowledge and Information Systems*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *Bioinformatics*, *The Modern Language Journal*, SDM, IEEE BigData, ICDM, CIKM, ECML-PKDD, DSAA, and AAAI. He has been invited to serve as the ad hoc reviewer and PC member of *Technology in Cancer Research & Treatment*, *Information Sciences*, *IEEE Transactions on Medical Imaging*, *IEEE Transactions on Fuzzy Systems*, *ACM Transactions on Knowledge Discovery from Data*, *IEEE Transactions on Green Communications and Networking*, *IEEE Internet of Things Journal*, *IEEE Transactions on Computational Social Systems*, *IEEE Transactions on Neural Networks and Learning Systems*, *TV*, *Journal of Machine Learning and Cybernetics*, ICDM, SDM, AAAI, CIKM, IJCAI, BIBM, PAKDD, KDD, FAccT, SMC, ICPR, ECML-PKDD, and ECAI. His personal website: https://www2.cs.siu.edu/~zchen/

Di Wu (Member, IEEE) received the PhD degree from the Chongqing Institute of Green and Intelligent Technology (CIGIT), Chinese Academy of Sciences (CAS), China, in 2019, and then joined CIGIT, CAS, China. He is currently a professor with the College of Computer and Information Science, Southwest University, Chongqing, China. He has more than 80 peer-reviewed publications, including 22 IEEE/ACM Transactions papers on *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Systems, Man, and Cybernetics*, and *ACM/IEEE Transactions on Information Systems*, and several conference papers on IEEE ICDM, AAAI, WWW, ECML-PKDD, IJCAI, etc. His research interests include machine learning and data mining. He is serving as an associate editor for *Neurocomputing* and *Frontiers in Neurorobotics*. His homepage: https://wudi1989.github.io/Homepage/